提出日：2024 年 5 月 17 日

先進情報プロジェクト演習

テーマ IT・1 第 4 回レポート

学籍番号：C0A22113
氏名：成田彩華

第１章　　CGI プログラムにおけるデータファイルの設置場所

第１回講義で作成した「/var/www/html/test/bbs.cgi」について考える。以下の URL にローカル環境からアクセスし、掲示板のプログラムが正しく動作していることを確認する。

http://192.168.64.7/test/bbs.cgi

5 行目が以下のようになっていることを確認する。

```
$DATAFILE = ‘./log.txt’;
```

次のような URL でアクセスするとどうなるか確認する。

http://192.168.64.7/test/log.txt

この URL にアクセスするとログファイルの中身が表示される。しかし、このように露ファイルが見れてしまうことはセキュリティ的に問題である。

/var/www/html/test に index.html があることを確認する。ない場合には、/var/www/html にある index.html をコピーして作成する。次のような URL でアクセスするとどうなるかを確認する。

http://192.168.64.7/test/

この URL にアクセスすると index.html が表示される。ここで、index.html を別の名前に変え、同じ URL にアクセスするとどうなるのか確認する。index.html が存在しない場合、/var/www/html/test/内のファイル一覧を表示させる。

http://192.168.64.7/test/log.txt でログファイルにアクセスできることが知られてしまったのかを考える。log.txt がある場所はドキュメントルートから辿れる test 直下にある。もし、index.html が存在しない場合、フォルダの中身が見れてしまうため、ログファイルがあることが知られてしまう。パーミッションを適切に設定していない場合は、log.txt を直接閲覧することが可能であると教えてしまっているようなものである。

/var/www/html/test のパーミッションを変更する。www-data に対して、「--x」になるよう設定する。設定したのち、以下の URL にアクセスする。

http://192.168.64.7/test/

アクセスすると Forbidden You don’t have permission to access this resource.と言われる。これは、パーミッション設定をしたことにより、このファイルに対して「ls」コマンドを実行する権限がなくなってしまったためである。しかし、実行権限はあるため、「cd」コマンドは実行できる。

パーミッション変更後の安全性を考えていく。ログファイルがルートディレクトリ上にある場合、パーミッション変更後でも「cd」コマンドは有効であるため、ログファイルにアクセスできてしまう。そのため、安全とはいえない。

第２章　　CGI プログラムにおけるデータファイルの適切な設置場所

bbs.cgi の log.txt を、ユーザのホームディレクトリの下に持ってくる。ユーザのホームディレクトリの確認は、以下のコマンドで確認できる。

```
$ cd
$ pwd
/home/ayaka
```

このように、ホームディレクトリが/home/ayaka/であることがわかる。ここに、/var/www/html/test/log.txt を mv コマンドで移動させる。

```
$ mv /var/www/html/test/log.txt /home/ayaka/
```

移動させた log.txt を読み込めるようにするため、bbs.cgi を書き換える。5 行目を以下のように書き換える。

```
$DATAFILE ‘/home/ayaka/log.txt’;
```

log.txt パーミッションを考える。このファイルはログファイルであるため、読み書きができる必要がある。読み書きするのは、www-data であるため、www-data に対して、「rw-」になるようにパーミッションを設定する。

第３章　　予約システムの PHP を動かす

講義ページから reservation.php と reservation_main.php をダウンロードし、/var/www/html/mail に設置する。この 2 つのファイルのパーミッションを考える。この 2 つのファイルは www-data が読み込める必要があるため、パーミッションは www-data「r--」になるよう設定する。

第４章　　Postfix のインストール

以下のコマンドを入力して Postfix のインストールを行う。

```
$ sudo apt install postfix
```

「Please select the mail server configuration type that best meets your nees.(あなたの用途にあったメールサーバ設定形式を選んでください。)」という画面が表示されたら、Tab キーで「<了解>」を選択する。

「General mail configuration type(メール設定の一般形式)」は「インターネットサイト」にしておく。

「System mail name(システムメール名)」は、書かれているまま(vm-ayaka)にしておく。

第５章　　PHP からのメール送信

reservation.php を reservation2.php として複製する。また、reservation_main.php を reservation_main2.php として複製する。

```
$ cp /var/www/html/mail/reservation.php /var/www/html/mail/reservation2.php
$ cp /var/www/html/mail/reservation_main.php /var/www/html/mail/reservation_main2.php
```

reservation2.php が呼び出すものが reservation_main2.php になるようにする。36 行目を以下のように変更する。

```
header('Location:reservation_main2.php');
```

reservation_main2.php の 6 行目の「$address = _SESSION['address'];」の下に、次の内容を追加して、予約内容が送信されるようにする。「$to」はその Linux の root にする。

```
$to      = 'root@localhost';
$subject = 'reservation';
$message = 'reservation:' . $user_name . ', ' . $phone . ', ' . $address;
$headers = array(
    'From' => 'ayaka@vm--ayaka',
    'Reply-To' => 'ayaka@vm-ayaka',
    'X-Mailer' => 'PHP/' . phpversion()
);

mail($to, $subject, $message, $headers);
```

第６章　　メールが届いたかの確認

http://192.168.64.7/mail/reservation2.php にアクセスし、メールを送信する。/var/mail/ 内に root というファイルがあるか確認する。

```
$ sudo cat root
```

以下のような内容のファイルが確認できる。

```
From www-data@vm-ayaka  Fri May 10 17:51:24 2024
Return-Path: <www-data@vm-ayaka>
X-Original-To: root@localhost
Delivered-To: root@localhost
Received: by vm-ayaka (Postfix, from userid 33)
. id 5E6EAE8569; Fri, 10 May 2024 17:51:24 +0900 (JST)
To: root@localhost
Subject: reservation
```

```
From: ayaka@vm-ayaka
Reply-To: ayaka@vm-ayaka
X-Mailer: PHP/8.1.2-1ubuntu2.17
Message-Id: <20240510085124.5E6EAE8569@vm-ayaka>
Date: Fri, 10 May 2024 17:51:24 +0900 (JST)

reservation:090, user, tokyo
```

第７章　　PHP からのメールの外部への送信

reservation.php を reservation3.php として複製する。また、reservation_main.php を reservation_main3.php として複製する。

```
$ cp /var/www/html/mail/reservation.php /var/www/html/mail/reservation3.php
$ cp /var/www/html/mail/reservation_main.php /var/www/html/mail/reservation_main3.php
```

reservation3.php が呼び出すものが reservation_main3.php になるようにする。36 行目を以下のように変更する。

```
header('Location:reservation_main3.php');
```

reservation_main3.php の 6 行目「$address = $_SESSION['address'];」の下に以下の内容を追加する。「$to」には自分の Gmail のメールアドレスを書く。

```
$to      = 'c0a221130b@edu.teu.ac.jp';
$subject = 'reservation';
$message = 'reservation:' . $user_name . ', ' . $phone . ', ' . $address;
$headers = array(
    'From' => 'ayaka@vm-ayaka',
    'Reply-To' => 'ayaka@vm-ayaka',
    'X-Mailer' => 'PHP/' . phpversion()
);

mail($to, $subject, $message, $headers);
```

第８章　　外部にメールが届いたかの確認

Gmial を確認する。以下のようなメールが届いていることが確認できる。

ayaka@vm-ayaka
To 自分

reservation:a, 090, tokyo

第 9 章　　tcpdump を使って Web サーバへの通信内容を見る

tcpdump のオプションについて理解する。「-A」はキャプチャデータの ASCII 表示。「src host」で送信元ホスト、「dst host」で送信先ホストを指定できる。「scr port」で送信元ポート、「dsp port」で送信先ポートを指定できる。

Web サーバへの通信内容を見る。reservation2.php から情報を送信してみる。以下のコマンドで通信内容を確認していく。

```
$ sudo tcpdump -A dst port 80 and dst host 192.168.64.7
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on enp0s1, link-type EN10MB (Ethernet), snapshot length 262144 bytes
21:01:29.341342 IP _gateway.tfido > vm-ayaka.http: Flags [SEW], seq 2872049110, win
65535, options [mss 1460,nop,wscale 6,nop,nop,TS val 3098238113 ecr 0,sackOK,eol],
length 0
E..@....@.y_..@...@....P./...........-.............
..¥.........
21:01:29.341816 IP _gateway.tfido > vm-ayaka.http: Flags [.], ack 72050365, win 2058,
options [nop,nop,TS val 3098238113 ecr 1660110805], length 0
E..4....@.yk..@...@....P./...Kf....
.......
..¥.b.G.
21:01:29.342161 IP _gateway.tfido > vm-ayaka.http: Flags [P.], seq 0:569, ack 1, win
2058, options [nop,nop,TS val 3098238114 ecr 1660110805], length 569: HTTP: POST
/mail/reservation2.php HTTP/1.1
E..m....@.w0..@...@....P./...Kf....
.......
..¥.b.G.POST /mail/reservation2.php HTTP/1.1
Host: 192.168.64.7
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
```

```
Accept-Encoding: gzip, deflate
Accept-Language: ja
Content-Type: application/x-www-form-urlencoded
Origin: http://192.168.64.7
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/605.1.15
(KHTML, like Gecko) Version/17.4.1 Safari/605.1.15
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Referer: http://192.168.64.7/mail/reservation2.php
Content-Length: 38
Cookie: PHPSESSID=287r4ctqan5b45cg9qo148169p



21:01:29.342556 IP _gateway.tfido > vm-ayaka.http: Flags [P.], seq 569:607, ack 1, win
2058, options [nop,nop,TS val 3098238114 ecr 1660110805], length 38: HTTP
E..Z....@.yC..@...@....P.0...Kf....
.......
..¥.b.G.user_name=user&phone=090&address=tokyo
21:01:29.344705 IP _gateway.tfido > vm-ayaka.http: Flags [.], ack 350, win 2053,
options [nop,nop,TS val 3098238117 ecr 1660110808], length 0
E..4....@.yk..@...@....P.0.6.Kh............
..¥.b.G.
21:01:29.349316 IP _gateway.tfido > vm-ayaka.http: Flags [P.], seq 607:1082, ack 350,
win 2053, options [nop,nop,TS val 3098238122 ecr 1660110808], length 475: HTTP: GET
/mail/reservation_main2.php HTTP/1.1
E.......@.w...@...@....P.0.6.Kh............
..¥.b.G.GET /mail/reservation_main2.php HTTP/1.1
Host: 192.168.64.7
Cookie: PHPSESSID=287r4ctqan5b45cg9qo148169p
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/605.1.15
(KHTML, like Gecko) Version/17.4.1 Safari/605.1.15
Referer: http://192.168.64.7/mail/reservation2.php
Accept-Encoding: gzip, deflate
```

```
Accept-Language: ja



21:01:29.367577 IP _gateway.tfido > vm-ayaka.http: Flags [.], ack 854, win 2045,
options [nop,nop,TS val 3098238140 ecr 1660110831], length 0
E..4....@.yk..@...@....P.0...Kj............
..¥.b.G.
```

この通信内容のうち、reservation2.php から送信した場合について、キャプチャした内容のうち、Web ブラウザから入力されて送信された内容は以下の通りである。

```
21:01:29.342556 IP _gateway.tfido > vm-ayaka.http: Flags [P.], seq 569:607, ack 1, win
2058, options [nop,nop,TS val 3098238114 ecr 1660110805], length 38: HTTP
E..Z....@.yC..@...@....P.0...Kf....
.......
..¥.b.G.user_name=user&phone=090&address=tokyo
```

「/var/www/html/test」に作成した、bbs.cgi から送信した場合について、キャプチャした内容は次の通りである。

```
$ sudo tcpdump -A dst port 80 and dst host 192.168.64.7
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on enp0s1, link-type EN10MB (Ethernet), snapshot length 262144 bytes
03:15:01.235080 IP _gateway.60999 > vm-ayaka.http: Flags [SEW], seq 3026577746, win
65535, options [mss 1460,nop,wscale 6,nop,nop,TS val 3344115281 ecr 0,sackOK,eol],
length 0
E..@....@.y_..@...@..G.P.e.R.....................
.S&Q........
03:15:01.235487 IP _gateway.60999 > vm-ayaka.http: Flags [.], ack 4055495302, win
2058, options [nop,nop,TS val 3344115281 ecr 1664753489], length 0
E..4....@.yk..@...@..G.P.e.S.......
.......
.S&Qc:.Q
03:15:01.235789 IP _gateway.60999 > vm-ayaka.http: Flags [P.], seq 0:551, ack 1, win
2058, options [nop,nop,TS val 3344115281 ecr 1664753489], length 551: HTTP: POST
/test/bbs.cgi HTTP/1.1
E..[....@.wB..@...@..G.P.e.S.......
k'.....
.S&Qc:.QPOST /test/bbs.cgi HTTP/1.1
Host: 192.168.64.7
```

```
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: ja
Content-Type: application/x-www-form-urlencoded
Origin: http://192.168.64.7
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/605.1.15
(KHTML, like Gecko) Version/17.4.1 Safari/605.1.15
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Referer: http://192.168.64.7/test/bbs.cgi
Content-Length: 32
Cookie: PHPSESSID=287r4ctqan5b45cg9qo148169p


03:15:01.236186 IP _gateway.60999 > vm-ayaka.http: Flags [P.], seq 551:583, ack 1, win
2058, options [nop,nop,TS val 3344115282 ecr 1664753489], length 32: HTTP
E..T....@.yI..@...@..G.P.e.z.......
.......
.S&Rc:.Qtitle=test&author=user&text=test
03:15:01.242197 IP _gateway.60999 > vm-ayaka.http: Flags [.], ack 889, win 2045,
options [nop,nop,TS val 3344115288 ecr 1664753496], length 0
E..4....@.yk..@...@..G.P.e.................
.S&Xc:.X
^C
5 packets captured
5 packets received by filter
0 packets dropped by kernel
```

このキャプチャ内容のうち、Web ブラウザから入力されて送信された内容は以下の通りである。

```
03:15:01.236186 IP _gateway.60999 > vm-ayaka.http: Flags [P.], seq 551:583, ack 1, win
2058, options [nop,nop,TS val 3344115282 ecr 1664753489], length 32: HTTP
E..T....@.yI..@...@..G.P.e.z.......
.......
.S&Rc:.Qtitle=test&author=user&text=test
```

第１０章　tcpdump を使って SMTP への通信内容をみる

reservation2.php から情報を送信してみる。以下のコマンドを打ち込みパケットをキャプチャする。

```
$ sudo tcpdump -A
```

メールに関して、それらしき内容が確認できない。Web サーバが SMTP に対応していないため、メールに関してのパケットが発生しない。

reservation3.php から送信した場合について、以下にメールに関してのキャプチャ内容の一部を示す。

```
$ sudo tcpdump -A


14:28:20.136225 IP vm-ayaka.41462 > tc-in-f27.1e100.net.smtp: Flags [P.], seq 1:16,
ack 74, win 502, options [nop,nop,TS val 3707512711 ecr 1754749980], length 15: SMTP:
EHLO vm-ayaka
E..C..@.@.r...@..........wu..*&....8......
..'.h.¥.EHLO vm-ayaka


14:28:20.306291 IP tc-in-f27.1e100.net.smtp > vm-ayaka.41462: Flags [P.], seq 74:241,
ack 16, win 256, options [nop,nop,TS val 1754750146 ecr 3707512711], length 167: SMTP:
250-mx.google.com at your service, [163.215.6.1]
E....}..x.........@.......*&..w...........
h.¥...'.250-mx.google.com at your service, [163.215.6.1]
250-SIZE 157286400
250-8BITMIME
250-STARTTLS
250-ENHANCEDSTATUSCODES
250-PIPELINING
250-CHUNKING
250 SMTPUTF8
```

reservation3.php から送信した場合、メールの本文がキャプチャできないのは、SMTP で送る際、google のサーバで暗号化通信が行われているため、通信内容自体が見れないためである。

PHP による送信方法が安全でない理由について考える。SMTP で送る際、いくつものサーバを経由する。経由するサーバが SMTP に対応していない場合、暗号化されずに通信が行われてしまう。また、SMTP にサーバが対応していても、経由するサーバに悪意のある何者かが侵入していた場合、通信内容を抜かれてしまう。そのため、PHP によるこの送信方

法が安全でない。

第１１章　ディレクトリ・トラバーサルを試す

まず、「/var/www/html/dirtra」に「view.php」を作成し、「/var/www/html/mail」にある「reservation.php」の内容をコピーする。

「user_name」を「item_name」に変更し、「ユーザ名」を「アイテム名」に変更する。ユーザ名以外の電話番号と住所についてはプログラム上から削除し、「アイテム名」に身について入力して送信されるようにする。送信先を「reservation_main.php」から「view_main.php」に変更し、HTML のタイトルも「予約」から「アイテム表示」に変更する。送信先を「reservation_main.php」から「view_main.php」に変更し、HTML のタイトルも「予約」から「アイテム表示」に変更する。変更後の view.php は以下の通りである。

```
<?php
ini_set('display_errors', true);
error_reporting(E_ALL);

function h($string)
{
    return htmlspecialchars($string, ENT_QUOTES, 'utf-8');
}

session_start();

// エラーを格納する変数
$err = [];

// 「ログイン」ボタンが押されて、POST通信のとき
if (filter_input(INPUT_SERVER, 'REQUEST_METHOD') === 'POST') {
    $item_name = filter_input(INPUT_POST, 'item_name');

    if ($item_name === '') {
        $err['item_name'] = 'アイテム名は入力必須です。';
    }
    // エラーがないとき
    if (count($err) === 0) {
        $_SESSION['item_name'] = $item_name;
```

```
        header('Location:view_main.php');
        return;
    }
}
?>
<!DOCTYPE HTML>
<html lang="ja">
    <head>
        <meta charset="UTF-8">
        <title>アイテム表示</title>
        <style type="text/css">
            .error {
                color: red;
            }
        </style>
    </head>
    <body>
        <div id="wrapper">
            <form action="" method="post">
                <p>
                    <label for="">アイテム名</label>
                    <input id="item_name" name="item_name" type="text" />
                    <?php if (isset($err['item_name'])) : ?>
                        <p class="error"><?php echo h($err['item_name']); ?></p>
                    <?php endif; ?>
                </p>

                <p>
                    <button type="submit">送信</button>
                </p>
            </form>
        </div>
    </body>
</html>
```

次に「/var/www/html/dirtra」に「view_main.php」を作成し、「/var/www/html/mail」にある「reservation_main.php」の内容をコピーする。「$_SESSION[‘item_name’]」として

「view.php」から渡されてくる内容を「$item_name」に格納するように変更する。

```
$item_name = $_SESSION[‘item_name’];
```

「$data_dir」という変数に、データが格納される「data」というディレクトリを相対パスで指定する。

```
$data_dir = ‘./data/’;
```

ファイルの内容を読み込むには次のようにすれば良い

```
$content = file_get_contents($data_dir . $item_name);
```

ファイルが存在しなかった場合には「$content」に「FALSE」が返されるので、次のようにしてファイルがなかったことを示すようにする。

```
if ($content === false) {
    $content = ‘None’;
}
```

以上のようにして「view_main.php」が出力する HTML の内容には、「view.php」で指定されたファイルの内容が表示されるようにする。以下に変更後の「view.php」を示す。

```
<?php
session_start();

$item_name = $_SESSION['item_name'];
$data_dir = './data/';
$content = file_get_contents($data_dir . $item_name);

if ($content === false) {
    $content = 'None';
}

?>
<!DOCTYPE HTML>
<html lang="ja">
    <head>
        <meta charset="UTF-8">
        <title></title>
    </head>
    <body>
        <p><?php echo $content; ?></p>
    </body>
</html>
```

データも作成する。「/var/www/html/dirtra/data」ディレクトリを作成し、その中に「dog」、「cat」、「mouse」という 3 つのテキストファイルを作成する。「dog」というファイルには「Dog」と書いておく。「cat」というファイルには「Cat」と書いておく。「mouse」というファイルには「Mouse」と書いておく。これらのファイルが Apach(www-data)からの読み込みアクセスができるようパーミッションを www-data に対して「r--」にしておく。

プログラムが完成したら、「view.php」で「dog」、「cat」、「mouse」、「tiger」を入力し、それぞれ何が表示されるか確認する。「dog」と入力すると「dog」ファイルの中身の「Dog」、「cat」と入力すると「cat」ファイルの中身の「Cat」、「mouse」と入力すると「Mouse」、が表示される。「tiger」と入力すると、ファイルが存在しないため「None」と表示される。

最後に、ホームディレクトリに設置した「log.txt」に Apach がアクセスできることを確認する。www-data に対して「r--」以上のパーミッションが設定されていることを確認する。その状態で、「view.php」にディレクトリ・トラバーサルを行い、「log.txt」の内容を表示できるか確認する。確認するにはアイテム名に../../../../../home/ayaka/log.txt を入力する。相対パスでアクセスするパスを指定しているため、このように入力することで、ホームディレクトリ上の log.txt にアクセスできる。

第１２章　PHP から HTML メールを送信する。

reservation.php を reservation4.php として複製する。reservation_main.php を reservation_main4.php として複製する。

予約情報が Web サーバに送信されると、予約内容をユーザ名と同名のファイルから読み込み、その内容を HTML メールとして Gmail に送信する。

reservation4.php の送信先を reservation_main4.php に変更する。reservation_main4.php の内容を以下のようにする。

```
<?php
session_start();


$user_name = $_SESSION['user_name'];
$data_dir = './data/';
$content = file_get_contents($data_dir . $user_name . '.txt');
if ($content === false) {
    $phone = $_SESSION['phone'];
    $address = $_SESSION['address'];
    file_put_contents($data_dir . $user_name . '.txt', $phone . "¥n" . $address .
"¥n");
}
```

```
else {
    $phone = $_SESSION['phone'];
    $address = $_SESSION['address'];
    file_put_contents($data_dir . $user_name . '.txt', $phone . "¥n" . $address .
"¥n");
    $fp = fopen($data_dir . $user_name . '.txt', 'r');
    $phone = fgets($fp);
    $address = fgets($fp);
    fclose($fp);
}



$to      = 'c0a221130b@edu.teu.ac.jp';
$subject = 'reservation';
$message = "<html lang='ja'><body><h1>complete reservation</h1><p>name: " . $user_name
. "</p><p>phone: " . $phone . "</p><p>address: " . $address . "</p><p>" . $content .
"</p>";
$headers = array(
    'From' => 'ayaka@vm-ayaka',
    'Reply-To' => 'ayaka@vm-ayaka',
    'Content-type' => 'text/html; charset=UTF-8'
);

mail($to, $subject, $message, $headers);
?>
<!DOCTYPE HTML>
<html lang="ja">
    <head>
        <meta charset="UTF-8">
        <title>¥u4e88¥u7d04¥u5b8c¥u4e86</title>
    </head>
    <body>
        <h>¥u4e88¥u7d04¥u5b8c¥u4e86</h>
        <p><?php echo $user_name; ?></p>
        <p><?php echo $phone; ?></p>
        <p><?php echo $address;?></p>
```

```
            <p><?php echo $content;?></p>
        </body>
    </html>
```

上記の通り、mail ディレクトリ直下にデータ保存用のディレクトリ「data」を作成し、そのディレクトリ内にファイルを保存する。ユーザ名が存在していなければ新たに「ユーザ名.txt」のファイルを作成し、入力した内容を書き込んでからそのファイルを読み込む。存在していれば新たに入力した内容に書き換え、読み込む。

HTML メールを送るには header で'Content-type' => 'text/html; charset=UTF-8'を指定することで HTML メールを送れるようになる。message を HTML で書いてあげることで HTML メールになる。

ホームディレクトリにある「log.txt」の内容もユーザ名を../../../../../home/ayaka/log と指定することで、相対パスを辿り、log.txt を表示させることができる。